

In the Specification

The specification has been amended as follows:

f1

Amend the paragraph beginning at page 6, line 24 as follows:

In a related aspect, the present invention provides a utility program program that defines new source code to be added to the existing source code.

f2

Amend the paragraph beginning at page 6, line 26 as follows:

In another related aspect, the present invention provides a utility program program that prioritizes the target code to update while employing the search technique.

f3

Amend the paragraph beginning at page 7, line 1 as follows:

In yet another related aspect, the present invention provides a utility program program that includes a process procedure for an operator to call, the process procedure recursively invokes the utility program and arguments.

f4

Amend the paragraph beginning at page 7, line 6 as follows:

In a related aspect, the present invention provides a utility program program that provides that the existing code with a specific character are not considered to be a file, and thereby are bypassed for any changes by the utility program.

Amend the paragraph beginning at page 8, line 18 as follows:

In describing the preferred embodiment of the present invention, reference will be made herein to ~~Figs. 1-3~~ Figs. 1-4 of the drawings in which like numerals refer to like features of the invention. Features of the invention are not necessarily shown to scale in the drawings.

Amend the paragraph beginning at page 8, line 22 as follows:

The present invention discloses using target patterns to provide a concise way to specify entire classes of dependencies. This method can replace many concrete (non-pattern-specific) rules in a build control file such as the *makefile* disclosed in the prior art. Other prior art methods use algorithms to build a complete concrete dependency tree before the utility starts. The present invention builds a pattern-based dependency tree instead, which can be done in minimal time without operating system calls. Further, the present invention determines which files match the patterns ~~as needed~~ as needed, thus saving time and computer resources.

Amend the paragraph beginning at page 16, line 17 as follows:

When target patterns are used in *upd^t* control files, there are occasions when it is helpful to be able to specify the dependency and update rules ~~associated with~~ associated with a given file separately.

Amend the paragraph beginning at page 17, line 1 as follows:

The first rule says that any file in the inst/bin directory depends on the corresponding file in the master/bin directory. The second rule says that any needed file in the inst/bin with a ".e" suffix ~~can be create~~can be created via a set substitution on its prerequisite. The third rule says that any needed file in the inst/bin should be simple a copy of its prerequisite. Because it comes after the ".e" rule, however, it does not get triggered for inst/bin/*.e files. It is only triggered by inst/bin files without a ".e" suffix.

8